DOCUMENT RESUME

ED 305 906 IR 013 763

AUTHOR	Brandt, Richard C.; Davis, Bradley N.
TITLE	Computer-Based Educational Software System. Final Report.
INSTITUTION	Utah Univ., Salt Lake City. Dept. of Computer Science.
SPONS AGENCY	Darbick Instructional Software Systems, Salt Lake City, UT.; Naval Personnel Research and Development Lab., Washington, D.C.
PUB DATE	13 Sep 88
CONTRACT	N00244-83-C-1759
NOTE	19p.; For a related report, see IR 013 762.
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Authoring Aids (Programing); *Computer Assisted Instruction; *Computer Games; Computer Graphics; *Computer Managed Instruction; Computer Simulation; Computer System Design; *Courseware; *Databases
IDENTIFIERS	*Computer Based Educational Software System; Computer Based Memorization System

ABSTRACT

CBESS (Computer-Based Educational Software System) is a set of 22 programs addressing authoring, instructional delivery, and instructional management. The programs are divided into five groups: (1) Computer-Based Memorization System (CBMS), which helps students acquire and maintain declarative (factual) knowledge (11 programs); (2) Language Skills Computer Aided Instruction (LSCAI), which helps students learn new vocabulary by presenting exercises of four to nine words (2 programs); (3) the Equipment Problem Solving Trainer (EPST), which permits authors without programming skills to create simulations of equipment (4 programs); (4) Script-Based Computer Assisted Instruction (SBCAI), which presents script-based instruction that creates "ad hoc" computer assisted instruction (3 programs); and (5) Instructional Management (2 programs). The two management programs are "Cop," which presents a linear sequence of lessons to a set of students, and "Course," which specifies the sequence of lessons, the mastery criteria, and student rosters. CBESS databases may be developed and delivered in both the MS-DOS and Berkeley UNIX environments. In addition, all CBESS programs except EPST may be run on the Novell network in protected mode. (7 figures and 24 references) (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED305906

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Final Report
Computer-Based Educational Software* System

Richard C. Brandt
Computer Science Department
University of Utah
Salt Lake City, Utah, 84112

Bradley N. Davis
Darbick Instructional Software Systems
P.O. Box 81157
Salt Lake City, Utah, 84108

September 13, 1988

The Computer-Based Educational Software System (CBESS) is an set of programs developed for the Navy Personnel Research and Development Center. It consists of 22 programs addressing authoring, instructional delivery, and instructional management. The programs are divided into five groups:

- Computer-Based Memorization System (CBMS) programs (11 programs)
- Equipment Problem Solving Trainer (EPST) programs (4 programs)
- Language Skills CAI (LSCAI) programs (2 programs)
- Script-Based CAI (SBCAI) programs (3 programs)
- Instructional Management (2 programs)

This report describes the CBESS programs and the instructional strategies built into them.

*This work was supported in part until April 1987 by the Navy Personnel Research and Development Center Contract N00244-83-C-1759. Since April 1987 this work has been supported by Darbick Instructional Software Systems.

1 CBMS

The Computer-Based Memorization System (CBMS) is a set of programs designed to help students acquire and maintain declarative (factual) knowledge. In addition, the programs can help students learn to:

- recognize items from graphics or video
- recall properties of items
- recall properties of categories
- discriminate among items
- identify items from descriptions
- select discriminating questions

CBMS consists of eight computer-based games, a program to browse through CBMS databases, a program to compile databases, and a program to decompile databases. The games are scored to make playing them interesting. In addition, a file containing an overlay model of the student's knowledge is built as the student plays the games. This file permits the games to adapt

IR013463

to a student's evolving knowledge. For example, when data in this file shows that a student knows a particular fact, the games stop asking questions about it.

CBMS uses a taxonomic hierarchy[1] to organize the items being described.[2] The CBMS semantic network supports:

- multiple inheritance
- attribution
- exceptions

CBMS also permits authors to bind to network nodes:

- pictures
- alternative answers

The games display the pictures of items where appropriate. They will match a student response against a list of acceptable alternatives if it doesn't match the node name. Because the alternative answers bound to a node are used only when the node's name is an expected answer they have only a local effect.[3] The alternative answers are not displayed to students.

[1]The semantic network used by CBMS is a standard knowledge representation and described in many places. For example, see the October 1986 issue (Vol. 74, No. 10) of the Proceedings of the IEEE.

[2]CBMS differs from standard semantic networks in that it does not distinguish between classes, categorized with the *IsA* relation and individuals, normally categorized with the *InstanceOf* relation. CBMS only uses *IsA*.

[3]The relation *SYNONYM_FOR* permits authors to create global alternatives. The answer parser substitutes global alternatives in its attempts to match student answers.

To simplify the semantic network, sentence generation, and search strategies, the facts are limited to three part facts, consisting of a subject, a relation, and an object. Examples of three part facts are:

```
"Ronald Reagan" "was elected" president;
California capital Sacramento;
abducens innervates "lateral rectus"
```

1.1 CBMS Games

In the CBMS games a distinction is made between recall and recognition. Usually students are first given the opportunity to type (recall) the answer. If students are unable to recall it they can switch to multiple choice and then attempt to recognize the answer in a list of possible answers. Students receive lower scores for recognition answers than for recall answers. Randomization and scoring make the games interesting and fun. The games provide students with the repeated exposure needed to memorize facts.

1.1.1 Concentration

Concentration's purpose is to reinforce the *perception of relationships* among items. The game uses a matrix game board in which each row and column correspond to items in a selected category and their attributes. Students choose a position on the board and answer a question that asks if the corresponding row and column items are related.

1.1.2 Constraint

Constraint's purpose is to help students learn to *discriminate among members* of a selected category. A statement and a list of category members are displayed and students mark the items in the list for which the statement is true.

1.1.3 Flashcard

Flashcard's purpose is to help students *recall all facts* about members of a selected category. Flashcard is an electronic version of conventional study aids based on flash cards.

1.1.4 Identify

Identify's purposes are to help students (1) *identify members* of a selected category from short descriptions and (2) *review facts* about the identified members.

1.1.5 Jeopardy

Jeopardy uses a matrix game board to teach students *to identify members* of selected categories from their complete descriptions.

1.1.6 Matrix

Matrix's purpose is to help students learn those items in the database *that have similar properties.* Matrix uses a matrix game board in which each row corresponds to the subjects of statements and each column corresponds to the relation portion of statements. When students select a box from the matrix, a question is presented asking for the objects that correctly complete a statement constructed using the row subject and the column relation.

1.1.7 Picture Quiz

Picture Quiz's purpose is to provide students with practice in recognizing objects from pictures.

1.1.8 Twenty Questions

Twenty Questions' purpose is to give students practice in *selecting and forming discriminating questions.* Students ask yes/no questions to identify an item selected by the game.

1.2 CBMS Authoring

There are two parts to authoring a CBMS database. The ASCII text file specifying the semantic network is created and edited with ordinary text editors. The sequences (pictures) associated with nodes are created and edited with the sequence editor, **se**. Sequences may include graphics, video, and text.

In addition to the statements specifying the semantic network, the text file includes:

- question templates specifying question formats

- alternative answer specifications
- display specification that identify sequences to associate with nodes

The text file and associated sequence files are compiled with the program `fromtext` to create the CBMS databases used with the games. The program `totext` decompiles CBMS databases to recreate the original text files.

1.3 CBMS and Memory

The design of CBMS was influenced by recent experimental and theoretical work on memory. John R. Anderson's spreading activation theory of memory as described in his book *Architecture of Cognition*[2] and other "semantic" memory models[23] provide the theoretical basis for CBMS. The unproven but plausible hypothesis in CBMS is that through repetitive use of the games, students may adopt the semantic organization that an author or authors have built into a database.

Pirolli and Anderson[18] studied the role of practice on fact retrieval. They found that practice spaced over many days is more effective in terms of the number of required repetitions for a given reaction time than extensive practice on a few days (see their Figure 3, p. 147). The CBMS games use a "two-in-a-row" rule to limit the number of repetitions in a single session with a game. The two-in-a-row rule stops the presentation of a particular question once a student has answered it correctly twice consecutively in a game.[4]

Authors may specify the minimum number of game sessions in which a student may answer a question involving a particular fact or item by specifying the "Importance" (a small number) of the fact or item. Each time a student answers a question involving a particular item or fact correctly, one is subtracted from its Importance; each time a student answers a question incorrectly, one is added to its Importance. The probability that a particular item or fact will appear in a question is proportional to its Importance. When the Importance of an item or fact reaches zero, it no longer appears in questions. The two-in-a-row rule keeps the Importance of an item or fact from going down more than two during any game session. This allows an author to specify in advance the minimum number of practice sessions that will include a particular item or fact. For example, if an author initializes the Importance of an item or fact at 10, the students will see questions involving the item or fact in at least five ($5 = 10/2$) separate sessions.

2 LSCAI

Language Skills Computer Aided Instruction (LSCAI) is designed to help students learn new vocabulary by presenting exercises of four to nine words. Since LSCAI addresses only vocabulary acquisition, its name is a misnomer, retained for historical reasons. LSCAI consists of two programs: `lsstu`, the instructional delivery program, and `lsaut`, the authoring program.

[4]The data of Pirolli and Anderson show that it is most efficient to correctly answer a question just once in a game. However, the students with whom we have worked, preferred getting the answer correct twice.

Lsstu follows generally accepted rules for vocabulary acquisition.[5] After the words are introduced either in a context that permits students *to deduce the meanings* of the words (from-context learning) or with *key words* and definitions (keyword learning),[6] student knowledge of the words is tested and reinforced in the following activities.

- Definition Building
- Hidden Multiple Choice
- Visible Multiple Choice
- Antonyms and Synonyms
- Spelling
- Definition Matching
- Labeling
- Cloze Paragraph

The activities can be divided into two groups.

- Individual Word Activities
- Collective Word Activities

Students should not do all the activities at once. Individual exercises may be specified from the operating system command line. For example, to do only the hidden multiple choice and labeling activities for the exercise "climate" in a database called "geography," a student would enter or an instructor would specify in a course file:

```
lsstu -hl geography climate
```

2.1 Individual Word Activities

The individual word activities concentrate on one word at a time. If a sequence of individual word activities is specified, for example if the five individual word activities are specified,

```
lsstu -dhvas geography climate
```

all five activities will be presented for one word before any activities are presented for the next word.[7]

2.1.1 Definition Building

Students "build" the word's definition by selecting phrases from a menu. If a selected phrase is wrong, LSCAI shows the number of the correct phrase and puts the correct phrase into the definition. The process continues until the definition is complete.

[5] There has been less reported research on vocabulary acquisition than on memory. The design of LSCAI was influenced more by prescriptions than by experimental data. Professors Gabe Della-Piana and Ralph Reynolds of the College of Education of the University of Utah provided the initial prescriptions.

[6] When LSCAI was designed, from-context learning of vocabulary words was the accepted method. It is still a widely accepted approach[8, 22]. However, recent experimental work by Dempster[7] and McDaniel et al[12, 19] has shown that other approaches, particularly keyword training, are more efficient. The sequence currently used to present the context material can be used without any software changes for keyword learning.

[7] Presenting all five individual word activities as illustrated in the command is *not* recommended. Dempster's experiments on spacing[7] have shown that it is better to "space" the activities for a word.

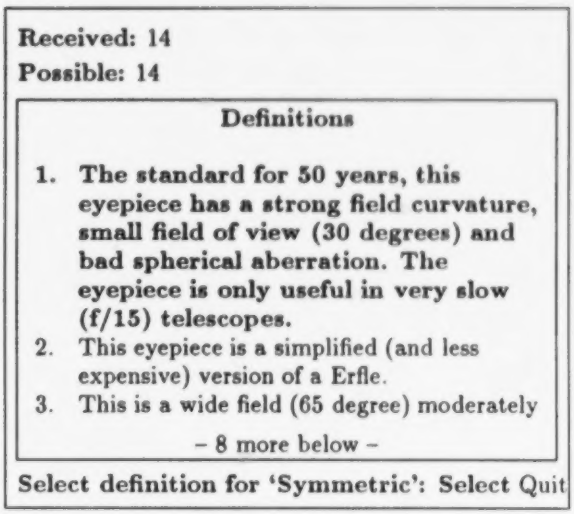
Received: 14
Possible: 14
Definitions
1. The standard for 50 years, this eyepiece has a strong field curvature, small field of view (30 degrees) and bad spherical aberration. The eyepiece is only useful in very slow (f/15) telescopes.
2. This eyepiece is a simplified (and less expensive) version of a Erfle.
3. This is a wide field (65 degree) moderately
– 8 more below –
Select definition for 'Symmetric': Select Quit

Figure 1: Visible Multiple Choice

2.1.2 Hidden Multiple Choice

Students view complete definitions, one at a time. They decide whether the displayed definition is the correct one. Definitions are displayed until either the correct definition for the current word is shown or a wrong definition is chosen. This activity is similar to the "semantic decision task" used by McKeown et al[13].

2.1.3 Visible Multiple Choice

Students choose the correct definition for the current word from a list of definitions. Figure 1, above, shows an example of the Visible Multiple Choice menu. This activity is similar to the multiple choice activity on most vocabulary exams.

2.1.4 Antonyms and Synonyms

Students identify words that are antonyms, synonyms, or related words for the word being studied from a list that also includes distractor words. This activity is skipped if there are no associated words.

2.1.5 Spelling

Students study a displayed word and press the *Enter* key when they are ready to spell it. The word is then erased and students may enter it.

2.2 Collective Word Activities

The collective word activities involve most of the words in an exercise.

Matching Students match the words shown in one menu with their paraphrased definitions shown in another menu. All the words in an exercise are displayed. Figure 2, p. 7, shows an example of a Matching activity menu.

Labeling A Sequence containing graphics, text, and video associated with exercise words is presented. Students enter the names of items in answer spaces next to the items.

Cloze Paragraph Students enter missing words in answer spaces to complete one or two displayed paragraphs.

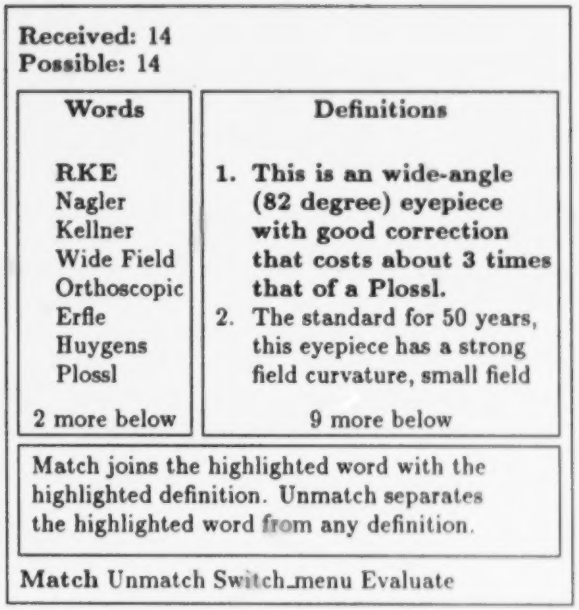
Received: 14
Possible: 14
Words
RKE
Nagler
Kellner
Wide Field
Orthoscopic
Erfle
Huygens
Plossl
2 more below
Definitions
1. This is an wide-angle (82 degree) eyepiece with good correction that costs about 3 times that of a Plossl.
2. The standard for 50 years, this eyepiece has a strong field curvature, small field
9 more below
Match joins the highlighted word with the highlighted definition. Unmatch separates the highlighted word from any definition.
Match Unmatch Switch_menu Evaluate

Figure 2: Matching Activity

2.3 LSCAI Authoring

Although LSCAI uses the same form of semantic network as CBMS, a special program, `lsaut`, is used for entering and editing word data, exercise data, and Sequences.

2.3.1 Word Data

Entry of word data is a form of fill-in-the-blank authoring. The components of word data and their use are listed below.

- From-context definition (Definition Building)
- Full definition (Multiple Choice)
- Paraphrased definition (Matching)
- Related context Sequence (On Error)
- Associated words (Antonyms and Synonyms)
- Confused words (Multiple Choice)

A word may have several sets of definitions if it has more than one meaning.

2.3.2 Exercise Data

Entry of exercise data is a form of fill-in-the-blank authoring. The components of exercise data and their use are listed below.

- Word list (defines the exercise)
- Context Sequence (Introduction)
- Sequence (Labeling)
- Label data (Labeling)
- Cloze paragraph (Cloze)

2.4 LSCAI and Vocabulary Acquisition

Vocabulary acquisition is significantly more complicated than memorization of facts. For example, the answers to the questions

```
What city is the capital of Maine?
Of which state is Augusta the capital?
```

are well defined when compared with the answers to the following two definition questions.

```
What is the capital of a state?
What serves as the seat of a state
 government?
```

These second two questions also show that there are two aspects to vocabulary knowledge; the knowledge of the meaning of a word (comprehension) and the converse issue of knowing the word that conveys a desired meaning (expression).

2.4.1 Comprehension and LSCAI

There are several stages in students' knowledge of word meanings[6].

1. Unrecognized. The word or phrase has not been seen before.
2. Recognized. The word or phrase has been seen before but its meaning is unknown.
3. Knowing in Context. The approximate meaning of the word or phrase is known only if the word is presented in context.
4. Definition Selectable. The correct definition of a word or phrase can be selected from a list of possible definitions.
5. Word Definition Recognizable. The correct definition of a word is recognized when it is presented by itself.
6. Word Definition Recall. The correct definition of a word can be recalled.

LSCAI can be used to get students to the fourth and fifth stages. The visible multiple choice and matching activities provide students with practice in definition selection and the hidden multiple choice activity provides students with practice in definition recognition. Although the multiple choice activities along with the associated word activity prepare students for the most common form of vocabulary test, the multiple choice test, their use in instruction has been criticized[9]. Consequently, they probably should be used only in combination with one or more of the other activities. For example, one way to introduce students to a set of words would be to present the context sequence followed by the visible multiple choice activity.

The contents of the context sequence introducing the words and of the related context sequences providing additional information when students choose the wrong definitions are controversial. Many educators believe that students should learn words from context rather than be presented with the definition because students learn most words from context. However, in a series of experiments, the psychologists McDaniel and Pressley[11] have shown that both the keyword method and the definition presentation method are more effective than the context method. Furthermore, they[12] have shown that the long term retention of definitions learned with the keyword method is the same as the long term retention of the definitions learned with the less efficient (it takes more trials to meet criterion) from-context method. Thus it now appears that the context sequence and the related context sequences should provide the student[8] with a keyword and

[8] McDaniel and Pressley have shown that the keyword method is most effective with low ability students (in their test students with verbal SAT scores less than 560); it was no more effective than the control of presenting just the definitions with the high ability students (verbal SAT scores greater than 560).

keyword illustration that relates the word to its definition where possible.[9]

There have been no experiments on vocabulary acquisition equivalent to Pirolli and Anderson's on memory[18]. However, it seems likely that their results would appiy to vocabulary acquisition as well. Working on new vocabulary words over a period of days is likely to more effective than concentrating the work on a single day. Beck et al[3] emphasize the importance of providing instruction over a period of days (5 in their case). In addition, Dempster's experiments[7] have demonstrated the value of "spacing" in a particular lesson. Consequently, it appears that LSCAI will be most effective if each group of exercise words is presented over a period of days, using only one or two of the activities each day.

2.4.2 Expression and LSCAI

There are several stages to a student's use of a word in written and oral expression.

1. Word Unknown. The correct word for a particular meaning is unknown.

2. Word Selectable. The correct word for a particular meaning (definition) can be selected from a list of possible words.

3. Word Recognizable. The correct word for a particular application can be selected from a list of possible words.

4. Word Recall. The correct word for a particular application can be recalled.

LSCAI can be used to get students to the third and fourth levels. The labeling and cloze activities require students either to recognize or recall words. Specifically, if a list of possible words are provided in the labeling and cloze activities, these activities provide students with practice in word recognition. If the list of words is omitted, the two activities provide students with practice in word recall.

3 EPST

The Equipment Problem Solving Trainer (EPST) was designed to permit authors without programming skills to create simulations of equipment. The simulations are used to provide students with the opportunity to observe how pieces of equipment work and to practice troubleshooting strategies. EPST was designed for practice; it provides no feedback to students aside from preventing them from taking dangerous actions and comparing the cost of their solution to the cost of an optimal solution.

Authoring in EPST is easy. It consists of specifying the behavior of individual (prototypical) parts and then connecting instances of parts together to build equipment. EPST's design was oriented toward digital systems; these systems are constructed by connecting together "off-the-shelf" parts. EPST supports the development

[9]The keyword must be acoustically related to the word to be defined. An example of a word keyword pair[19] is "angler" and "angel." The illustration is an "angel who is fishing." It is often not possible to find an appropriate keyword and illustration, particularly when the word has a precise technical meaning.

of libraries of prototypical parts. The use of libraries has both advantages and drawbacks. The primary advantage is that once a prototypical part and its behavior have been specified in an EPST library it can be used in many pieces of equipment without any additional effort. The primary disadvantage is that a part's output cannot depend on the properties of the parts to which it is connected because these are unknown until instances of the part are connected to other parts. This precludes the use of EPST for systems where a part's output depends on the devices to which it is connected. For example, EPST cannot be used for analog circuits or analog process simulations such as PLANT[17].[10]

A part's state is recomputed whenever one of its inputs changes. Production rules are used to determine the new state. The rule conditions may involve the part's inputs, previous state, and error conditions. For each part, state functions are used to specify its outputs in terms of its inputs, current state, and error conditions. The part data is kept in databases containing only part information.

Equipment is specified in separate databases containing the connectivity information, equipment specific data (initialization data is usually equipment specific), and instances[4] of the parts. Each equipment database is bound to the database containing the part information because the part data is not copied into the equipment database. The part instances in the equipment databases contain the dynamical data (input, output, state, and error values) and data identifying the prototypical part in the part database.

EPST consists of four programs: a student program, `epstu`, an authoring program, `epaut`, a type editing program, `eptype`, and a problem specification program, `praut`.

3.1 EPST Student Program

`Epstu` permits students to:

- learn how equipment works
- practice trouble-shooting techniques

To make the practice realistic, video images of the actual equipment overlaid with graphics to represent the equipment and its current state may be used. Alternatively, graphics images symbolizing the equipment and its components may be used.[11] An example of an EPST graphics image is shown in Figure 3, p. 11.

Students use the EPST commands shown in Figure 3 to:

- obtain more detailed views of parts of

[10] EPST may be used in some analog circuits by adding "hidden" devices, "dæmons," that do the necessary computations. Authors must design and specify the performance of these "dæmons." For example, an electrical voltage drop "constraint dæmon" might compute the current flow through a group of parts from the driving voltage(s) and the resistances of the parts. These "dæmons" depend on the connectivity and subvert the original goal of EPST that the parts be independent of the connectivity. In addition, "invisible" wires must be added to the system to transmit the information needed by the "dæmons."

[11] If it is impossible to display video or graphics images, EPST will display menus containing data representing the current condition of the equipment.

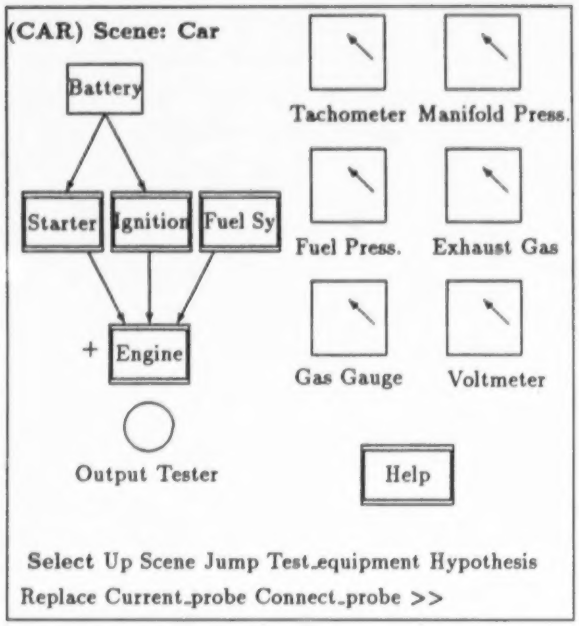

Figure 3: Example EPST Screen

equipment. For example, if students select Ignition in Figure 3, the screen will be cleared and graphics representing the components of the ignition will be displayed.

- observe displays that reflect the state of the equipment. For example, in Figure 3, the appearance of the Tachometer, Fuel Pressure, Gas Gauge, Manifold Pressure, Exhaust Gas Temperature, and Voltmeter gauges all reflect the state of the car.
- make measurements. Students may attach probes to testpoints and obtain readings. The probes may be attached either to test equipment or to the equipment under test.
- change switch settings,
- replace parts of the equipment, and
- enter hypotheses.

Students start the problem-solving session by reading the "complaint." The complaint is a written description of one or more symptoms of the problem. The objective in EPST is to find and replace the faulty part.

3.2 EPST Authoring

Three programs, `epaut`, `eptype`, and `praut` are used for EPST authoring.

`Epaut` is used to:

- create and edit the EPST databases
- specify how a piece of equipment is subdivided into components
- specify how components of the equipment work
- specify how the component parts of the equipment are connected together
- identify positions at which students may take measurements (testpoints)

The information that authors enter using `epaut` determines how the equipment will "work" when students use the database.

`Epaut` is used to enter assertions such as:

1. device A is a D flip flop
2. cable 17 is a RS-232 cable
3. cable 17 is connected to the terminal's connector B

epaut is also used to enter rules such as:

1. if the Clear input is asserted then a D flip flop will enter state 0
2. if the Reset input is asserted then a D flip flop will enter state 1
3. if the clock doesn't rise then a flip flop will not change state

Actual rules follow a standard form. For example, the second rule above has the form: (the comments in italics and the rule has been displayed on several lines for clarity).

```
If in Any_State The state doesn't matter
and 0  The part is not faulted
and (Reset = Low) Reset is asserted
then 1 The D flip flop will go to the On state
```

By entering both assertions and rules, authors can describe how something (a piece of equipment, an organization, or an organism) works.

Eptype is used to edit *data types.* When authors specify a data type they specify the values the data may have and how the data will be represented in the computer. For example, a data type called electric_light could have the values on, off, and broken.

Praut is used to specify the problems in a piece of equipment. When authors specify a problem they provide a problem name, solutions, a brief description of the problem (the "complaint"), and they identify the faulty part and the type of fault (a part can usually be faulted in several different ways).

3.3 EPST and Trouble Shooting

EPST was designed to serve as a replacement of the Generalized Maintenance Trainer Simulator (GMTS), now known as the Electronic Equipment Maintenance Trainer (EEMT)[10]. Practice on both GMTS[20] and EEMT[5] was as effective as practice on the actual equipment.[12]

It is difficult to teach people to be good troubleshooters. Morris and Rouse[15] have summarized the research on troubleshooting. They found that while practice was useful, theoretical instruction (p. 513) was ineffective. In a later experiment[16] with the PLANT simulator, they found that in achieving the primary goal of maximizing the PLANT production, the performances of four test groups were identical. All test groups received minimal instruction. The control group received no additional instruction. One group received procedural instruction, another received instruction on the principle, and a third received instruction on both principles and procedures. The groups receiving procedural instruction did control PLANT in a "more stable manner" (p. 704).

It is possible that practice with EPST may help students develop better knowledge of the topology of the system and the effects of faults. Because the appearance of parts in EPST is up to the authors, authors can present more information to students than is provided by the actual

[12] Both GMTS and EEMT use a touchscreen and videodisc images of the actual equipment to achieve a high degree of physical fidelity. EPST supports the use of text and graphics as well as video and consequently can be used to achieve a much lower degree of physical fidelity.

equipment.[13] For example, a part's output may be reflected in the color of the part. Sometimes this will trivialize the problem. For example, if the AND nodes in TASK One[21] with output 0 were assigned one color and those with output 1 were given another color, the source of the fault would be obvious. However, working or playing with such a system might help students develop "schema" or "frames" involving the effects of fault propagation, particularly if they are given the opportunity to insert faults.[14] In a slightly more complicated system (a synchronous counting circuit using gates and JK flip flops), we associated colors with parts' states. We have found that it is not trivial to locate the faults in this system, even though the parts' states are displayed. We also found that sometimes it is impossible to determine the faulted part from state data. The display of parts' states relieves students of taking measurements and permits them to concentrate on the topology and the effects of fault propagation.

4 Script-Based CAI

Script-Based CAI (SBCAI) is used to present script-based instruction and consequently serves the same purpose as other script-based systems such as TENCOR, WISE, and PLATO. Like these other authoring systems, it is used to create "ad hoc" CAI. There is no built-in instructional strategy.

Because authors create databases rather than lessons with the SBCAI editor, the SBCAI systems forms a 4th Generation Language system. SBCAI databases are formed from five entities:

- Primitives. These provide for display of materials (graphics, video, text) in windows.
- Linear Sequences. Sequences are used for the sequential (no branching) display of primitives.
- Interactions. Interactions present expository material, a question, and appropriate local feedback material.
- Synonym frames. These specify the synonyms to be used in answer analysis.
- Control and Computation frames. These specify the flow of control in a lesson, using as data information returned from the Interactions and other Control and Computation frames. A Control and Computation frame may invoke other Control and Computation frames as well as any of the other entities. The control language is an easy-to-learn subset of Modula-2.

4.1 SBCAI Student Program

The SBCAI student program, `try`, permits students to use traditional script-based CAI. Students use `try` by specifying at the command-line the database and the entity or entities to be used. For example, to view material on the negation strategy, "negation" in a database on

[13] Most experimenters, for example, Allen[1], have addressed the opposite issue, the effects of reducing physical fidelity.

[14] In EPST, students have to use the authoring program `praut` to insert faults. This is to be contrasted with the "What If" support provided to operators in the "aiding" system developed by Yoon and Hammer[24]. In this system, operators can deliberately insert faults into a simulator to see their effects.

problem solving "problem," a student would enter or an instructor would specify in a course file:

```
try problem negation
```

The same database can be used for a variety of purposes. Because the specific purpose is accessed by specifying at the commandline the entity or entities to use, the same database can be used for the pretest, instruction, and posttest.

4.2 SBCAI Authoring

Authors use the Lesson and Interaction Editor, `lie`, to edit Interactions, Control and Computation frames, Synonym frames, and Sequences.

4.2.1 Interactions

An Interaction is used to:

- present sequences of text, graphics, and video frames
- present a question
- obtain and categorize an answer
- present an appropriate response sequence

All interaction with students in SBCAI is through Interactions, and each Interaction obtains the answer to a single question. The authoring involves "fill-in-the-slot" operations since an Interaction has a predefined structure. Although the structure is fixed, authors are given considerable flexibility because

Interaction: multiple_test

Question-Answer Type Combinations

•1. **Window – keyboard answer**
2. Line – keyboard answer
3. Commandline – single answer
4. Menu – single answer
5. Multiple Choice – single answer
6. Matching Question – vector answer
7. True-False Question – vector answer
8. Menu – vector answer
9. Multiple Choice – vector answer
10. Position – position answer

Select from Above

QUESTION ANSWERS: Select Quit

Figure 4: Question-Answer Type Menu

ten question-answer types are supported. The Question-Answer Type Menu is shown above. A predefined structure and the use of menus to show what has been entered permit authors to maintain an overview of the structure of an Interaction.

4.2.2 Control and Computation Frames

Control and Computation frames:

- present Interactions
- present other Control and Computation frames
- execute Synonym frames
- present Sequences
- compute variables used in a lesson

Control and Computation frames are edited in windows or copied from ordinary ASCII text files. Because Control and Computation frames use a programming language, they may be used to build a wide variety of lessons ranging from linear lockstep lessons to lessons offering a substantial degree of student control. The programming language is a subset of Modula-2.

4.2.3 Synonym Frames

A Synonym frame is a table of conversion rules to be used in analysis of student answers. Synonym frames free authors from specifying all the acceptable synonyms for each keyboard answer. For example, if the following synonym frame is executed,

```
km  kilometer  1.0
km  meter  1000.0
m  meter  1.0
m  yard  1.0936111
```

and in a later Interaction, the anticipated answer is 1500 meters, then student answers of 1.5 km, 1.5 kilometer, and 1640 yards will match. Synonym frames are edited in windows or they may be copied from ordinary ASCII text files.

4.2.4 Sequences

Sequences are used to display a linear sequence of primitives. All primitives in a Sequence are presented to students. When a Sequence is edited by an author, a menu like that shown in Figure 5, above, is displayed on the screen.

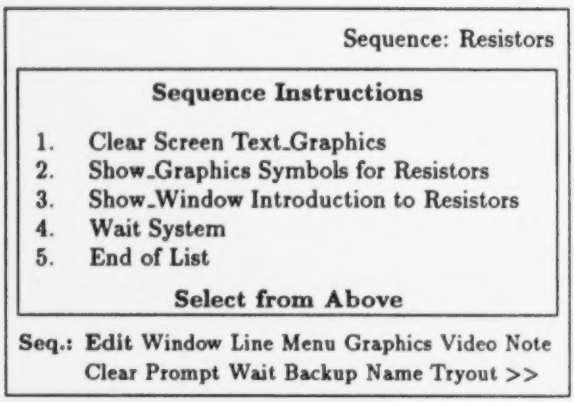
Sequence: Resistors
Sequence Instructions
1. Clear Screen Text_Graphics
2. Show_Graphics Symbols for Resistors
3. Show_Window Introduction to Resistors
4. Wait System
5. End of List
Select from Above
Seq.: Edit Window Line Menu Graphics Video Note
Clear Prompt Wait Backup Name Tryout >>

Figure 5: Sequence Instruction List

Sequences may be edited with both `lie` and the Sequence Editor, **se**.

Text The following four types of text primitives are supported:

- label primitives to place labels on the screen
- menu primitives to display a list of items
- line primitives to display text on a single line
- window primitives to display paragraphs of text and other freely formed text

The four types of text primitive are illustrated in Figure 6, above. Text presented in the window and menu primitives used in questions can be scrolled by students if not all the text can be displayed at once.

Graphics With **se** and `lie`, authors can create both graphics primitive and collections of

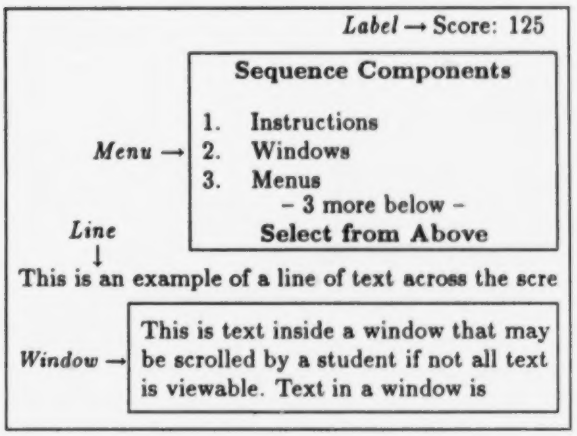
Label → Score: 125
Sequence Components
1. Instructions
2. Windows
3. Menus
– 3 more below –
Select from Above
Menu →
Line
This is an example of a line of text across the scre
Window →
This is text inside a window that may be scrolled by a student if not all text is viewable. Text in a window is

Figure 6: Text Primitives

reusable graphics primitives called graphics libraries. A graphics primitive is illustrated in Figure 7, p. 16.

Video The video presented to students is specified using video control primitives. Video control primitives contain only the information that specifies how to display video materials, that is, which frames, speed, forward or backward, etc. The actual video material resides on a videodisc.

5 Instructional Management

Two programs are used for instructional management: `cop` and `course`. `Cop` presents a linear sequence of lessons to a set of students. The sequence of lessons and the mastery criteria are specified in a course file while the student group is specified in a roster file. Students are prohibited from progressing to the next lesson in the sequence until they meet mastery criteria specified by the author or instructor. `Course` is used to specify the sequences of lessons, the mastery criteria, and student rosters.

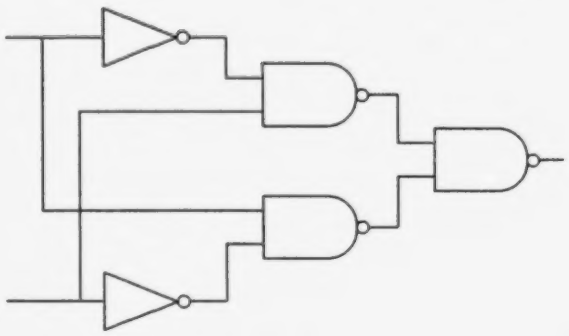

Figure 7: Graphics Frame

6 Development and Delivery

CBESS databases may be developed and delivered in both the MS-DOS[15] (versions 3.1 and higher) and Berkeley UNIX[16] (versions 4.2 and 4.3) environments. All CBESS programs run in both environments and datafiles created in the UNIX environment may be used in the MS-DOS environment and vice versa. In addition, all CBESS programs except EPST may be run on the Novell network in protected mode.

In the UNIX environment, only a text screen is supported and graphics is shown as text. In

[15] MS-DOS is a Trademark of Microsoft Inc.

[16] UNIX is a registered trademark of AT&T Bell Laboratories.

the MS-DOS environment, a range of display adapters are supported (mda, hercules, cgahi, cga80, cgalow, cga40, ega) along with two input devices (Mousesystems mouse and Summagraphics digitizer). Three brands (Hitachi, Sony, and Pioneer) of videodisc are supported. A file that is usually called `cbess.cfg`[17] specifies the devices to use. CBESS supports three logical screens: author, student, and overlay. Different display adapters can be bound to the logical screens; material specified for a particular logical screen will be displayed on the device connected to the display adapter to which it is bound. The CBESS programs were designed to run on systems with hard disks. They do not run as well on floppy disk systems.[18]

References

[1] John A. Allen, Robert T. Hays, and Louis C. Buffardi. Maintenance training simulator fidelity and individual differences in transfer of training. *Human Factors*, 28(5):497–509, October 1986.

[2] John R. Anderson. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, 1983.

[3] Isabel L. Beck, Margaret G. McKeown, and Richard C. Omanson. The effects and uses of diverse vocabulary instructional techniques. In McKeown and Curtis [14], chapter 9, pages 147–163.

[4] Ronald J. Brachman. What's in a concept: Structural foundations for semantic networks. *International Journal of Man-Machine Studies*, 9(2):127–152, March 1977.

[5] L.F. Cicchinelli, R.A. Keller, and K.R. Harmon. Training capabilities test of electronics equipment maintence trainer: Findings and conclusions. Draft report, Denver Research Institute, University of Denver, Denver, CO., February 1984.

[6] Mary E. Curtis. Vocabulary testing and vocabulary instruction. In McKeown and Curtis [14], chapter 3, pages 37–51.

[7] Frank N. Dempster. Effects of variable encoding and spaced presentations on vocabulary learning. *Journal of Educational Psychology*, 79(2):162–170, 1987.

[8] Patrica A. Drum and Bonnie C. Konopak. Learning word meanings from written context. In McKeown and Curtis [14], chapter 5, pages 73–87.

[9] Edward J. Kameenui, Robert C. Dixon, and Douglas W. Carnine. Issues in the design of vocabulary instruction. In McKeown and Curtis [14], chapter 3, pages 129–145.

[10] George F. Lahey and Vernon M. Malec. Generalized maintainance trainer simulator: User manual. Special Report NPRDC SR 82-17, Navy Personnel Research and Development Center, San Diego, CA 92152, March 1982.

[17] The name of the file is specified in the environment variable `CBESS`.

[18] Some of the programs will not run in the root directory. When these programs are run on floppy disk systems, they must be placed in a subdirectory and executed from there.

[11] Mark A. McDaniel and Michael Pressley. Putting the keyword method in context. *Journal of Educational Psychology*, 76(4):598–609, 1984.

[12] Mark A. McDaniel, Michael Pressley, and Paul K. Dunay. Long-term retention of vocabulary after keyword and context learning. *Journal of Educational Psychology*, 79(1):87–89, 1987.

[13] Margaret G. McKeown, Isabel L. Beck, Richard C. Omanson, and Martha T. Pople. Some effects of the nature and frequency of vocabulary instruction on the knowledge and use of words. *Reading Research Quarterly*, 20(5):522–535, 1985.

[14] Margaret G. McKeown and Mary E. Curtis, editors. *The Nature of Vocabulary Acquisition.* Lawrence Erlbaum Associates, Hillsdale, N.J, 07642, 1987.

[15] Nancy M. Morris and William B. Rouse. Review and evaluation of empirical research in trouble shooting. *Human Factors*, 27(5):503–530, October 1985.

[16] Nancy M. Morris, William B. Rouse, and Janet L. Fath. The effects of type of knowledge upon human problem solving in a process control task. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(6):698–707, November 1985.

[17] Nancy M. Morris, William B. Rouse, and Janet L. Fath. Plant: An experimental task for the study of human problem solving in process control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(6):792–798, November 1985.

[18] Peter L. Pirolli and John R. Anderson. The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(1):136–153, January 1985.

[19] Michael Pressley, Joel R. Levin, and Mark A. McDaniel. Remembering versus inferring what a word means: Mnemonic and contextual approaches. In McKeown and Curtis [14], chapter 7, pages 107–127.

[20] J.W. Rigney, D.M. Towne, J.P. Moran, and R.A. Mishler. Field evaluation of the generalized maintenance trainer-simulator. Technical Report 80-30, Navy Personnel Research and Development Center, 1980.

[21] William B. Rouse, Sandra H. Rouse, and Susan J. Pellegrino. A rule-based model of human problem solving performance in fault diagnosis tasks. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-10(7):366–376, July 1980.

[22] Robert J. Sternberg. Most vocabulary is learned from context. In McKeown and Curtis [14], chapter 6, pages 89–105.

[23] Steven Yantis and David E. Meyer. Dynamics of activation in semantic and episodic memory. *Journal of Experimental Psychology: General*, 117(2):130–147, 1988.

[24] Wan C. Yoon and John M. Hammer. Aiding the operator during novel fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-18(1):142–148, February 1988.